

Docket No.: POU920040027US1

Inventor: Paul A. Bunce et al.

For: **METHOD FOR SKIP OVER  
REDUNDANCY DECODE WITH  
VERY LOW OVERHEAD**

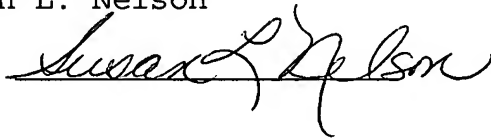
APPLICATION FOR UNITED STATES

LETTERS PATENT

"Express Mail" Mailing Label No. ER865633021US  
Date of Deposit: March 31, 2004

I hereby certify that this paper is being deposited with the United States Postal Service as "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: Box Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Name: Susan L. Nelson

Signature: 

INTERNATIONAL BUSINESS MACHINES CORPORATION

# **METHOD FOR SKIP OVER REDUNDANCY DECODE WITH VERY LOW OVERHEAD**

## **FIELD OF THE INVENTION**

**[0001]** This invention relates to standalone memory products such as SRAM, and DRAM and embedded memory applications such as Microprocessors, and particularly to a method of implementing I/O redundancy control logic with a minimal impact to both chip area and chip wire tracks..

## **DESCRIPTION OF BACKGROUND**

**[0002]** Redundancy is a technique that is used to replace a defective element of a chip with a spare element. At least two types of redundancy are used in RAM design. Wordlines have been traditionally repaired by a “real-time” compare of an address and a known defective address. If the address matches the known defective address, the main array is bypassed and a spared wordline is activated. Skip-Over Redundancy has been used to repair defective I/O circuits and bit columns. See the patent entitled “A SYSTEM FOR IMPLEMENTING A COLUMN REDUNDANCY SCHEME FOR ARRAYS WITH CONTROLS THAT SPAN MULTIPLE DATA BITS”, of Paul Bunce, John Davis, Thomas Knips, and Donald Plass, U.S. Patent 6,584,023, issued June 24, 2003 (which is incorporated herein by reference) which describes as an exemplary embodiment a system for implementing a column redundancy scheme for arrays with controls that span multiple data bits. The system includes an array of data bits for receiving data inputs, a spare data bit and a field control input line. Also included in the system is circuitry to separate a field control signal from the field control input line into one or more individual control signals for activating a corresponding data bit in the array

or for input to a multiplexor. The system further comprises circuitry to steer around a defective data bit in the array. This circuitry includes: a field control signal multiplexor corresponding to each field control signal; a spare control signal multiplexor to activate the spare data bit; a data multiplexor corresponding to each of the data bits in the array; and a spare data multiplexor to steer one of the data inputs

## **SUMMARY OF THE INVENTION**

**[0003]** The invention described below is a method of implementing I/O redundancy control logic with a minimal impact to both chip area and chip wire tracks. While our prior solution described in the referenced U.S. Patent 6,584,023 we could implement the decode and carry function as a centralized function. However, the solution required real estate usage on a chip that was undesirable. Our solution to this problem enables use of odd and even decoder outputs that can share a single wire track, the same wire being utilizable for both odd and even decoder outputs. Using the method described below, the odd and even decoder outputs can share the same wire track. This reduces the number of wire tracks from 1 track per I/O to 1 track per 2 I/Os. Additional features and advantages are realized through the techniques of the present invention. Other embodiments and aspects of the invention are described in detail herein and are considered a part of the claimed invention. For a better understanding of the invention with advantages and features, refer to the description and to the drawings.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

**[0004]** The subject matter which is regarded as the invention is particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other objects, features, and advantages of the invention are

apparent from the following detailed description taken in conjunction with the accompanying drawings in which:

- [0005] FIG. 1 illustrates one example of Skip Over Redundancy with no blocks skipped over.
- [0006] FIG. 2 illustrates one example of Skip Over Redundancy with block n skipped over.
- [0007] FIG. 3 illustrates one example of Skip Over Redundancy using decode and carry.
- [0008] FIG. 4 illustrates one example of a logical and transistor level implementation of decode and carry.
- [0009] FIG. 5 illustrates one example of a physical implementation that requires one vertical wire per block.
- [0010] FIG. 6 illustrates one example of a physical implementation that fits in macro white space.
- [0011] FIG. 7 illustrates one example of the odd/even decode scheme with the decode function separate from the carry function.
- [0012] FIG. 8 illustrates one example of a physical implementation of odd/even decode scheme with the decode function fitting into macro white space and requiring only one wire track per block.
- [0013] The detailed description explains the preferred embodiments of the invention, together with advantages and features, by way of example with reference to the drawings.

## DETAILED DESCRIPTION OF THE INVENTION

[0014] Turning now to our invention, our method uses a Skip-Over technique which requires a set of muxes at the input and output of a block that is to be repaired. Our improved method of implementing I/O redundancy control logic has a minimal impact to both chip area and chip wire tracks. Our solution to overcome problems of required real estate usage on a chip that was undesirable enables use of odd and even decoder outputs that can share a single wire track, the same wire being utilizable for both odd and even decoder outputs. In order to implement the decode and carry function as a centralized function, there arises a requirement that logically adjacent decode circuits (decoders connected by a carry signal) should be physically close together to minimize the overhead of the carry wiring. If the decode structure and the mux structure are arranged orthogonal to each other, then each decoder output would require a wire track. (as shown in Figs. 5 and 6). Using the method described below, the odd and even decoder outputs can share the same wire track. This reduces the number of wire tracks from 1 track per I/O to 1 track per 2 I/Os.

[0015] Fig. 1 and Fig. 2 demonstrate the Skip-Over technique. In Fig. 1 all the shift lines are low, indicating that no repair is required. In this case the redundant element is not used and the data flow from input to output follows the path traced by the dotted line. In Fig. 2 all the shift lines above block n are all high indicating that block n is to be repaired or “skipped-over”. The dotted lines show the path from input to output. Note that this time block n is “skipped-over” and spare element is used.

[0016] The shift control signals to these muxes can be generated by a decode and carry structure. Fig. 3 shows a Skip-Over implementation of I/O redundancy using a decode and carry structure to control the mux select signals. In this example the decoder inputs are programmed to select the mux to the right of the block that needs to be skipped over, the carry feature forces a select on all the mux select signals to the right of the programmed mux. The inputs to the decoder are typically “configured” before the data

path begins normal operation. This means that the outputs of the decode and carry structure are not time critical.

[0017] Fig. 4 is an implementation of the decode and carry circuit used in Fig. 3. The logic function is an “and-or”. The transistor level implementation is a natural choice for this logic function because it implements the function using a minimal number of transistors.

[0018] Fig. 5 shows an example of a RAM floorplan that uses the skip over technique to implement I/O redundancy. In this example, there are two sets of I/O circuits, one in the right half macro and one in the left half macro. Each half has a set of I/O circuits and a “spare”. The decode and carry structure is shared by both sets of I/O circuits. In this example the decode and carry structure is placed at the side of the RAM and the decode wires flow horizontally across the memory cells and then vertically down to the I/O circuits and muxes. There are two disadvantages with this implementation. First, the decode and carry occupies additional space outside the RAM. Secondly the implementation requires one vertical wire for each I/O.

[0019] Fig. 6 is an enhancement to the implementation. In this case, the decode and carry function is broken up into two groups; each group provides the mux controls for half of the I/O circuits. This allows the function to fit into “whitespace” in upper and lower half of the RAM, saving space. The number of vertical wires increases by one (the carry signal from the lower half to the upper half). The number of vertical wires driving to the muxes remains the same. Half the wires come from the bottom and half the wires come from the top.

[0020] Fig. 7 is an implementation of skip-over redundancy using the decode and carry function. In this case, however, instead of using the centralized decode and carry circuit shown in the left side of Fig. 4, the circuit is repartitioned. In this case the AND portion of the decode has been separated from the OR (carry) portion of the circuit. In this implementation the OR function is physically located with the mux and I/O circuit,

permitting the “odd” and “even” numbered decoders to be physically disjoint. In this example, the “even” decoders drive to the I/O circuits from above while the “odd” decoders drive from the bottom. Since the “odd” and “even” decoders are driving to the I/O circuits from different directions, and the I/O circuits they drive are physically adjacent it is possible to share the wire tracks (in this case the vertical tracks) between the “odd” and “even” decode wires.

[0021] Fig. 8 demonstrates a floorplan that implements this methodology. Separating the decode and carry logic, and partitioning the decode into “odd” and “even” groups allows the decode function to be broken up two or more blocks which are more easily placed into existing ‘whitespace’. Furthermore, the number of wire tracks required to implement the function is reduced by half.

[0022] The examples shown in this specification assume two sets of I/O circuits each with one repair each. In practice this methodology would apply to a wide range of RAM configurations, each with a wide range of repair actions.

[0023] While the preferred embodiment to the invention has been described, it will be understood that those skilled in the art, both now and in the future, may make various improvements and enhancements which fall within the scope of the claims which follow. These claims should be construed to maintain the proper protection fo the invention first described.